Course Syllabus

Jump to Today Edit

No profession unleashes the spirit of innovation like computer science and engineering. Computer software runs the modern world. In the last week, what was the longest stretch of time you went without interacting with a computer in some way? Think of cell phones, cars, Google, Facebook, Twitter, eBay, Wikipedia, Roomba, World of Warcraft, Amazon, and NOAA's hurricane prediction, most of which are less than a decade old. Moreover, computation is now radically transforming the social sciences and humanities. Students will benefit by being able to apply computational thinking to their own study and research areas.

About the Course

The objective of this course is not to turn you into an expert programmer. Rather, this course is designed to give you an overview of computer science and teach you about problem solving in a way that utilizes computation. Upon completion of this course, you will have a feel for how to think about and structure problems in such a way that you can use a computer to help you solve them. Programming is necessarily a part of this process, but it is neither the only part nor the most important part.

Organization

The course will be organized around the following main applications. You will then explore algorithms and implementations of computational solutions for those problems. These will introduce Python programming and computational problem solving throughout the course.

1. **Customizing House Plans:** Using architectural cost models.
2. **Rock-Paper-Scissors-Lizard-Spock:** Playing a game from popular culture.
3. **Statistics:** Solving statistical problems.
4. **Mathematical Modeling:** Modeling and exploring the behavior of populations in a predator-prey relationship as represented by Lotka-Volterra equations.
5. **Databases:** Storing, updating, and finding information in a large data set.
6. **Text Analysis:** Analyzing statistical properties of natural language texts. Generating similar natural language text automatically via Markov models.
7. **Social Networks:** Building social networks and measuring properties thereof.
8. **Machine Learning:** Using neural nets to recognize visual patterns.

Structure

Before class, you are expected to watch the video lectures and take the online quiz that correspond to the class. These materials and their schedule are listed on the Modules page. These quizzes are graded.

During class, you will work on exercises (see Class Schedule or Modules) that apply the ideas from the videos. You will work in groups. Participation during class -- i.e., attending and working on the exercises — counts as part of the course grade. Completing the exercises successfully is not required, but doing so will help on the assignments. Exercise solutions are provided after class. Sometimes the exercises are pieces of the assignments.

> **Summer session only:** Class is online via teleconferencing (see Conferences) at the scheduled class time — 10:30am-12n Central Time. If there is sufficient demand, I will also hold class in a classroom, so that Houston-based students can attend in person. Working in groups is optional for those teleconferencing. Due to scheduling issues, attendance is not required, but still strongly recommended.

Assignments are done outside class, although you can work on them during class if you've finished the class-time exercises. Each assignment has two parts — one where you will turn in code, and one where you will turn in text.

Exams will be during 3-hour scheduled periods outside of class. Multiple times will be offered to accommodate schedules.

Workload

The time that students spend varies tremendously, but the following numbers are what is reported by students.

- **Before each class**: About half hour of videos — sometimes more, sometimes less. Students skip through or re-watch as desired. On some devices, you might be able to speed up playback of the videos. A short quiz that students typically spend about a half hour on.

- **During class**: Class attendance and participation is expected. But, you always attend class, anyways, right?
- **Assignments**: Nine per semester — a week or two per assignment. Students average about 5 hours per assignment, but with a wide range of 1 to 15 hours per assignment.

Grading

Your course grade will be broken down as follows:

Activity	Description	Spring/Fall	Summer
Class attendance & participation	You are expected to attend class and work with a group on exercises. Completing the exercises successfully is not required.	10%	0%
Before-class quizzes	There will be a short quiz before each class. You will have three attempts at each quiz (with no penalty). You will be able to see the explanations/hints immediately after you submit.	10%	10%
Assignments	Nine assignments, completed individually.	50%	55%
Exams	Two exams, completed individually.	30%	35%

The code on each assignment will be partially auto-graded. When you turn it in, you'll see a score that is recorded as your grade. However, your actual score may go up, as the human graders will look at your code and assign partial credit to problems that you missed.

Late Submissions

The Canvas system has the notion of a "due date" and a "available until date". The "due date" is the real deadline by which you should turn in your work. The "available until date" is the time after which the system will stop accepting submissions.

Late penalty: We will apply a late penalty of 10 points per day for assignments and quizzes (not exams) turned in after the due date.

Late days: Each student is allowed 3 late days or automatic penalty-free 24-hour extensions of assignment deadlines (not quizzes or exams). These will be applied by the graders at the end of the semester without any required student action.

Extensions

Extensions allow students to turn in graded work late without a penalty. Extensions will only be granted under exceptional circumstances (such as medical emergencies). Having lots of work and deadlines is not an exceptional circumstance; it is part of being a college student.

Grade Disputes

If you believe your grade on an assignment is incorrect, you have **7 days** from when the assignment was returned to bring this to the attention of the staff.

1. Discuss your assignment with the original grader.
2. You can appeal to the TA in charge of grading (see the Course Staff in the navigation), or to the instructor during the summer.

Students with Disabilities

If you have a documented disability that will impact your work in this class, please contact me in the first two weeks of class to discuss your needs. Additionally, you will need to register with the Disability Support Services Office in the Allen Center.

Assignments Summary:

Date	Details	
Wed Aug 24, 2016	Quiz - Expressions and Variables (https://canvas.rice.edu/courses/942/assignments/8588)	due by 10am
Fri Aug 26, 2016	Quiz - Functions (https://canvas.rice.edu/courses/942/assignments/8595)	due by 10am
	Quiz - Expressions and Variables (https://canvas.rice.edu/courses/942/assignments/8588) (4 students)	due by 11:59pm

Date	Details	
Mon Aug 29, 2016	Quiz - Expressions and Variables (https://canvas.rice.edu/courses/942/assignments/8588) (2 students)	due by 10am
	Quiz - Functions (https://canvas.rice.edu/courses/942/assignments/8595) (3 students)	due by 10am
Wed Aug 31, 2016	Quiz - Logic and Comparisons (https://canvas.rice.edu/courses/942/assignments/8599)	due by 10am
Fri Sep 2, 2016	Assignment 1 Python - Customizing House Plans (https://canvas.rice.edu/courses/942/assignments/8605)	due by 10am
	Assignment 1 Text - Customizing House Plans (https://canvas.rice.edu/courses/942/assignments/8606)	due by 10am
Wed Sep 7, 2016	Quiz - Strings, Lists, Tuples (https://canvas.rice.edu/courses/942/assignments/8590)	due by 10am
Fri Sep 9, 2016	Quiz - For-Loops (https://canvas.rice.edu/courses/942/assignments/8597)	due by 10am
Mon Sep 12, 2016	Assignment 2 Python - RPSLS (https://canvas.rice.edu/courses/942/assignments/8607)	due by 10am
	Assignment 2 Text - RPSLS (https://canvas.rice.edu/courses/942/assignments/8608)	due by 10am
Wed Sep 14, 2016	Quiz - Random and Sequences (https://canvas.rice.edu/courses/942/assignments/8598)	due by 10am
Mon Sep 19, 2016	Assignment 3 Python - Statistics (https://canvas.rice.edu/courses/942/assignments/8609)	due by 10am
	Assignment 3 Text - Statistics (https://canvas.rice.edu/courses/942/assignments/8610)	due by 10am
	Quiz - Plotting, Zip (https://canvas.rice.edu/courses/942/assignments/8589)	due by 10am
Mon Sep 26, 2016	Assignment 4 Python - Predator-prey (https://canvas.rice.edu/courses/942/assignments/8611)	due by 10am
	Assignment 4 Text - Predator-Prey (https://canvas.rice.edu/courses/942/assignments/8612)	due by 10am
	Quiz - Dictionaries (https://canvas.rice.edu/courses/942/assignments/8592)	due by 10am
Wed Sep 28, 2016	Quiz - Dictionary Variations (https://canvas.rice.edu/courses/942/assignments/8594)	due by 10am
Mon Oct 3, 2016	Assignment 5 Python - Art Database (https://canvas.rice.edu/courses/942/assignments/8613)	due by 10am
	Assignment 5 Text - Art Database (https://canvas.rice.edu/courses/942/assignments/8614)	due by 10am
	Exam 1 (https://canvas.rice.edu/courses/942/assignments/8623)	due by 10pm
Wed Oct 5, 2016	Quiz - Processing File Data (https://canvas.rice.edu/courses/942/assignments/8601)	due by 10am
Wed Oct 12, 2016	Quiz - Regular Expressions 1 (https://canvas.rice.edu/courses/942/assignments/8602)	due by 10am
Fri Oct 14, 2016	Quiz - Markov Chains (https://canvas.rice.edu/courses/942/assignments/8591)	due by 10am
Mon Oct 17, 2016	Quiz - Regular Expressions 2 (https://canvas.rice.edu/courses/942/assignments/8604)	due by 10am
Fri Oct 21, 2016	Quiz - Nested Data Structures (https://canvas.rice.edu/courses/942/assignments/8600)	due by 10am

Date	Details	
Wed Oct 26, 2016	Assignment 6 Python - Text Analysis (https://canvas.rice.edu/courses/942/assignments/8615)	due by 10am
	Assignment 6 Text - Text Analysis (https://canvas.rice.edu/courses/942/assignments/8616)	due by 10am
	Quiz - Advanced Iteration (https://canvas.rice.edu/courses/942/assignments/8585)	due by 10am
Mon Oct 31, 2016	Quiz - Graphs and Sets (https://canvas.rice.edu/courses/942/assignments/8587)	due by 10am
Wed Nov 2, 2016	Assignment 7 Python - Text Riffing (https://canvas.rice.edu/courses/942/assignments/8617)	due by 10am
	Assignment 7 Text - Text Riffing (https://canvas.rice.edu/courses/942/assignments/8618)	due by 10am
	Quiz - Graphs and Object-Oriented Programming (https://canvas.rice.edu/courses/942/assignments/8596)	due by 10am
Mon Nov 7, 2016	Quiz - Function and Object Definitions (https://canvas.rice.edu/courses/942/assignments/8603)	due by 10am
Wed Nov 9, 2016	Quiz - Objects and Mutation (https://canvas.rice.edu/courses/942/assignments/8584)	due by 10am
Wed Nov 16, 2016	Assignment 8 Python - Social Graphs (https://canvas.rice.edu/courses/942/assignments/8619)	due by 10am
	Assignment 8 Text - Social Graphs (https://canvas.rice.edu/courses/942/assignments/8620)	due by 10am
	Quiz - Optimization (https://canvas.rice.edu/courses/942/assignments/8593)	due by 10am
Mon Nov 21, 2016	Quiz - Binary Numbers (https://canvas.rice.edu/courses/942/assignments/8586)	due by 10am
Fri Dec 2, 2016	Assignment 9 Python - Neural Nets (https://canvas.rice.edu/courses/942/assignments/8621)	due by 10am
	Assignment 9 Text - Neural Nets (https://canvas.rice.edu/courses/942/assignments/8622)	due by 10am
Wed Dec 7, 2016	Exam 2 (https://canvas.rice.edu/courses/942/assignments/8624)	due by 5pm
	Roll Call Attendance (https://canvas.rice.edu/courses/942/assignments/8625)	

Honor Code

We take the **Honor Code (http://honor.blogs.rice.edu/files/2011/01/Academic-Fraud-and-the-Honor-System.pdf)** very seriously. The work you submit for this class is expected to be the result of your own work. Attempting to take credit for someone else's work by turning it in as your own constitutes plagiarism, as defined by the Rice Honor Code.

Assignments

You may use any materials provided by the course. You may use your own materials, including code from your group's in-class exercises.

You may collaborate with anyone or with any website or book. However, you may use at most two lines of code from any source. Changing variable names or otherwise transforming code does not allow you to exceed that two-line rule. You should comment your code to cite any sources.

Quizzes

You may use any materials provided by the course. You may use your own materials, including code from your group's in-class exercises.

After you have taken the quiz once, you may discuss the contents and topics of the quiz as much as you like with others, but you should not share actual solutions.

Exams

You will need to bring a laptop to use Canvas, CodeSkulptor & its documentation, OwlTest, and CanvasTest. (I cannot guarantee power outlet availability.) I can bring a Chromebook for you to use, if you notify me ahead of time.

You may use the laptop to write and test code. You may access anything provided as part of this course, including your own notes and in-class group work. Bring headphones if you will access course videos. You may not access any other website.

You may read your written notes and write on scratch paper. You may not use any books.

No collaboration is allowed.

Advice and Resources

Helpful advice

The most important advice for almost any problem is **ASK** a question. Ask in class, in office hours, or on Piazza.

- Having trouble understanding CodeSkulptor's error messages? **Guide to Python Errors** (https://canvas.rice.edu/courses/332/pages/guide-to-python-errors)
- Having trouble understanding OwlTest/CanvasTest's error messages? **Guide to Pylint Errors** (https://canvas.rice.edu/courses/332/pages/guide-to-pylint-errors)
- Having trouble finding that bug in your program? Review the videos on debugging techniques, and use these ideas. Step through your code either mentally or with Python Tutor. Try finding the bug on your own before asking for help, as trouble-shooting is an important skill beyond just in programming.
- Having trouble understanding a video presentation? Try the online tutorials at the bottom of this page for alternate presentations.
- Having trouble finishing your assignments in a reasonable amount of time? There are lots of possible reasons for this. Truly participate during class and don't just listen to others solving exercises. Don't program by trial-and-error; follow the problem-solving strategies introduced in the course.
- How to best study for exams? Do exercises. We provide exam reviews with exercises. Also, go back and redo class exercises and assignment problems. Look at the online tutorials below for additional exercises.

Required text

- None
- **Coding Style and Standards** (https://canvas.rice.edu/courses/942/pages/coding-style-and-standards)
- **Writing Recipes** (https://canvas.rice.edu/courses/942/pages/writing-recipes)

Tools

- Required: **rice.codeskulptor.org** (http://rice.codeskulptor.org) — edit and save Python code
- Optional: **www.pythontutor.com** (http://www.pythontutor.com) — visualize the execution of Python code

Python references

- **Guide to Python Errors** (https://canvas.rice.edu/courses/942/pages/guide-to-python-errors)
- **Guide to Pylint Errors** (https://canvas.rice.edu/courses/942/pages/guide-to-pylint-errors) (i.e., what OwlTest/CanvasTest reports)

No specific manual is required. Searching via Google is often the most effective option, and the following are some good online sources:

- Official **Python 2.7 documentation** (http://docs.python.org/2.7/)
- **CodeSkulptor Documentation** (http://www.codeskulptor.org/docs.html) for the Python version that CodeSkulptor supports

For printed reference manuals, a few good options are

- **"Beginning Python"** (http://hetland.org/writing/beginning-python-2/) by Hetland
- **"Learning Python"** (http://shop.oreilly.com/product/9780596158071.do) by Lutz
- **"Introduction to Computing and Programming in Python"** (http://www.pearsonhighered.com/educator/product/Introduction-to-Computing-and-Programming-in-Python-A-Multimedia-Approach/9780131176553.page) by Guzdial

The following are useful online tutorials for Python:

- **"Official" Python Tutorial** (http://docs.python.org/tutorial)
- **Non-Programmer's Tutorial for Python 2.6** (http://en.wikibooks.org/wiki/Non-Programmer's_Tutorial_for_Python_2.6)
- **A Beginner's Python Tutorial** (http://www.sthurlow.com/python/)
- **ZetCode's Python tutorial** (http://zetcode.com/tutorials/pythontutorial/)

Course Staff

These office hours are not yet finalized. Office hours will start the second week of classes.

Name	Email	Office Hours		Classification	Role
John Greiner	greiner	appointment	Duncan Hall 3093	Faculty	Instructor
Bolun Li	bl24	appointment	Duncan Hall 3099	Grad	Grading Lead TA
Aditya Shrotri	as128	appointment	Duncan Hall 3060	Grad	Technical Lead TA
Bailey Brashears	bnb4	M 3-4pm	Hanszen Commons	Undergrad	Grader
Khang Nguyen	kdn1	M 4-5pm	Baker Commons	Undergrad	In-class TA (am), Grader
Rishi Suresh	rs49	M 6-7pm	Lovett Commons	Undergrad	In-class TA (pm), Grader
Shirley Jiang	xj2	T 1-2pm	Brochstein Pavilion	Undergrad	In-class TA (am), Grader
Martin Zhou	cz16	T 4-5pm	Baker Commons	Undergrad	Grader
Yi Fung	yrf1	T 8-9pm	Hanszen Commons	Undergrad	In-class TA (pm), Grader
Kevin Chang	ktc1	T 9-10pm	Lovett Commons	Undergrad	Grader
Jamie Hall	jmh12	W 11am-12n	Duncan Commons	Undergrad	In-class TA (pm), Grader
Chang Gao	cg25	W 3-4pm	Jones Commons	Undergrad	Grader
Ani Kunaparaju	avk3	W 6-7pm	Jones Commons	Undergrad	Grader
Matthew Holcombe	mch8	W 8-9pm	Duncan Commons	Undergrad	In-class TA (pm), Grader
Ziyi Ye	zy13	Th 11am-12n	Lovett Commons	Undergrad	Grader
Naomi Eigbe	nee1	Th 2:30-3:30pm	Jones Commons	Undergrad	In-class TA (am), Grader
Iris Gau	ig10	Th 7-8pm	Duncan Commons	Undergrad	Grader
Tiffany Stuart	tjs9	Th 8-9pm	Lovett Commons	Undergrad	In-class TA (pm), Grader
Swapnav Deka	sd55	Su 6:30-7:30pm	Sid Commons	Undergrad	Grader

When emailing staff, please include **COMP 130** in your subject line. Please use Piazza for questions of general interest.